IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2021, accepted November 28, 2021, date of publication December 3, 2021,
date of current version December 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3132684

Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data

WISDOM UDO AND YAR MUHAMMAD (Senior Member, IEEE)
Department of Computing and Games, School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, U.K.
Corresponding author: Wisdom Udo (wisdomudo213@gmail.com)

ABSTRACT Rystad Energy's analysis shows that installed offshore wind capacity will rise to 27.5 GW in 2026 from 10.5 GW in 2020. This report indicates that increasingly complex maintenance needs must be met for wind turbines(WTs). IRENA report shows that offshore wind operation and maintenance (O&M) costs typically constitute 16-25% of the cost of electricity for offshore wind farms deployed in the G20 countries. Data collection and analytics, predictive maintenance, and production output optimisation of WTs must be explored to increase operational reliability and reduce maintenance costs of WTs. Predictive maintenance in wind turbines can be achieved by analysing data obtained by sensors already equipped with the WT. This network of sensors forms part of a Supervisory Control and Data Acquisition (SCADA) system. We developed a method for monitoring and detecting anomalies in the WT critical components, such as the gearbox and the generator. The proposed approach is based on the historical SCADA data that is common in most wind farms. We developed models using extreme gradient boosting (XGBoost) and Long Short-Term Memory (LSTM) to build the characteristics behaviour of critical WT components, and Statistical Process Control (SPC) was used to evaluate its anomalous behaviour. The proposed method was tested on two real case studies regarding six different WT to determine its effectiveness and applicability.

INDEX TERMS Wind turbine, fault detection, supervisory control and data acquisition (SCADA), extreme gradient boosting (XGBoost), predictive maintenance, statistical process control (SPC), long short-term memory (LSTM).

I. INTRODUCTION

The government is committed to transit away from fossil fuels and decarbonising the power sector to eliminate contributions to climate change by 2050. In 2020, the UK generated 43.1% of its electricity from renewable sources, with the wind making up 24.2% [1]. Rystad Energy's analysis shows that installed offshore wind capacity is set to rise to 27.5 GW in 2026 from 10.5 GW in 2020 [2]. As shown by Rystad Energy analysis in Fig.1, the future trend is more of these wind turbines being installed in the offshore environments and less onshore [3]. This indicates that increased complex maintenance needs must be met for such equipment. IRENA report shows that offshore wind operation and maintenance (O&M) costs typically constitute 16-25% of the cost of electricity for offshore wind farms deployed in the G20 countries. To drive these costs, optimising O&M practices to reduce unscheduled maintenance needs to be unlocked by improvements in data collection and analytics, allowing for predictive maintenance and production output optimisation [4].

A Wind turbine is a device that captures wind energy through its rotating blades and converts the wind energy into electrical energy using its drivetrains. Wind turbine drivetrains are classed into direct drive(DD) and gear type, which has a gearbox; both classes have a hub as the input, the main shaft as the transfer and the generator as the output [5]. Other wind turbine components include main shaft bearings, mechanical brake, shaft bearing, yaw systems, power electronic systems, hydraulic and cooling systems. The gearbox and generator play critical roles in the energy conversion process from the WT components mentioned above. Since WT gearboxes operate in a high-altitude nacelle, to reduce the weight and enhance the transmission ratio, planetary transmission is widely adopted in WT gearboxes [6]. Therefore, WT gearboxes has been designed as a planetary/spur gearbox system where the spur gearbox is the fixed gearbox stage. The fixed gearbox stage increases the rotational speed of the planetary gear consequently leading to induced vibration manifesting as strong noise in the WT gearbox. Due to the

The associate editor coordinating the review of this manuscript and approving it for publication was Kostas Kolomvatsos.

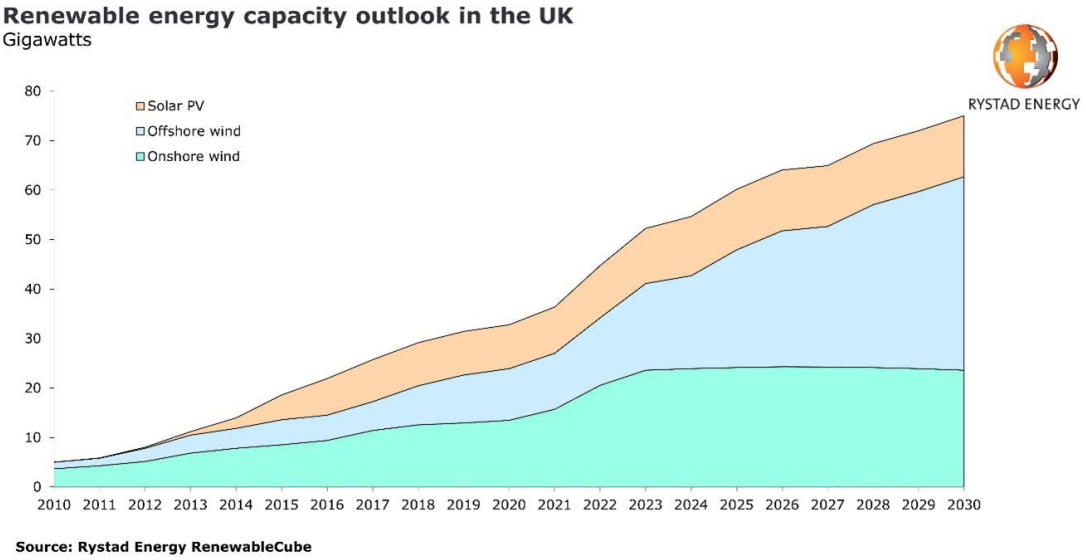


FIGURE 1. Rystad Energy analysis showing the future trend of more WT installed in the offshore environment.

stochastic nature of wind, the rotational speed is time varying making it difficult to diagnose fault in the WT gearbox system [7]. The kinetic energy of the wind is transformed into rotational energy by the shaft connected to the turbine's blade when the wind hits the blade. The moving shaft is connected to a generator, which produces electrical power through electromagnetism [3]. The double fed induction generator (DFIG) is extensively employed in gearbox driven WT, whose operation mode is based on the rotational speed of the rotor windings and stator windings connected to the transformer. The rotor windings are connected to the power grid through an inverter that regulates slip power based on the rotational speed of the rotor. The rotor sends power to the grid at ultra-synchronous speed, whereas the stator transfers all active power to the grid at the synchronous speed of the generator [8]. The rotating shaft of the generators when the rotational speed is lower than the synchronous speed of the generator, the rotor absorbs energy from the grid. It is mainly supported by bearings, which qualifies it as one of the most critical components in a WT. As the generator shaft continuously rotates, bearing damage may emerge, so effective fault detection is necessary. This raises concerns since it can be costly and dangerous to perform maintenance. WTs are often deployed in harsh environments and remote locations such as offshore environments to maximise wind motion. WTs can be hundreds of feet above the ground, requiring lifting maintenance crew with a crane or dropping them from a helicopter. Hence, the need to monitor what is going on with our equipment is necessary to avoid such dangerous and costly activities and perform maintenance when needed. Typically, organisations adopt various maintenance programs to increase operational reliability and decrease costs, and these programs can be reactive, preventive, or predictive. In reactive maintenance, the equipment is used to its limits, and repairs are performed when components have become defective. Preventive maintenance is also known as scheduled maintenance, and here maintenance is carried out at a regular rate to avoid failures. The challenge here is determining when to do maintenance since we do not know when failure will occur; hence organisations use a conservative approach in planning maintenance for safety-critical equipment. The problem here is that if maintenance is scheduled very early, this will waste machine life that is still useful, which adds to costs. If we can predict when failure will occur, we schedule maintenance right before it [9]. Predictive maintenance is performed based on condition monitoring (CM), a technique that informs maintenance of equipment and components that are likely to fail and have them replaced at the right time [10]. So predictive maintenance helps asset managers to bridge the gap between reactive maintenance and scheduled maintenance by carrying maintenance not too late or too early but just-in-time. Predictive maintenance can help us: estimate time to failure (remaining useful life), detect problems in our equipment (anomaly detection) and help us identify what parts need to be fixed (diagnosis of fault types). The challenge of predictive maintenance can be solved by first-principles modelling, that is, using a physics-based approach. This does not require any data coming off from the wind turbines but does require a large amount of domain expert knowledge. It involves deriving equations that tell us how the system behaves, and from that, we can use those equations to determine how the equipment will degrade and eventually fail over time. On the other hand, data-driven modelling does not require expert knowledge of the system evaluation but instead requires a good amount of data taken off the real-world system. We then use several statistical and machine learning techniques to develop models based on the data to help us understand the system behaviour and how it

fails [11]. There are also several hybrid approaches, where data-driven strategies are used fill to the knowledge gap about the first principles of the system.

Over the past decade, there has been a rapid increase in autonomous condition monitoring systems to monitor equipment performance, including wind turbines. Condition monitoring strategy can be applied based on the vibration-sensor system, which has vibration sensors, strain gauges, or oil particle counters retrofitted to turbine sub-components for localised monitoring [12]. The problem with this condition monitoring strategy is the cost involved in retrofitting the sensors and the data collection and analysis required to provide insight into system performance [13]. Wind turbines are equipped with sensors that records data of the equipment state, this network of sensors form part of a Supervisory Control and Data Acquisition (SCADA) system. The SCADA system obtained by the network of sensors was initially installed to monitor and operate the system. Still, recently this engineering data have been harnessed to identify anomalies and access the health status of the wind turbine, paving the way for data-driven predictive maintenance [14]. The sensors forming the SCADA system are in the main components of the wind turbine; the data is usually sampled at a frequency of 10-min. This sampling interval makes it easy for data transfer and storage in a database for ultimate retrieval [15]. SCADA Systems on a WT typically record wind parameters like wind speed and wind deviations; performance parameters like power output, rotor speed, blade pitch angle; vibration parameters like tower acceleration and drive train acceleration; temperature parameters like bearing temperature and gearbox temperature. All these recorded parameters could be used to perform fault detection and prognosis activities [16]. Capturing all this data will help us develop a robust algorithm that can better detect faults. This has created a SCADA system-based condition monitoring system when the captured data can be evaluated at different levels of granularity. At the most fine-grained level, we can monitor the condition of wind turbine sub-components such as drivetrain. Also, at the most coarse-grained level, we can monitor the whole wind turbine by combining signals of different components to provide a high-level warning [10]. When considering sub-components to monitor, decisions should be based on failure rates and downtime per failure. Priority is given to components that are more prone to failure and have extended lead times for replacement [17]. Data based on a survey of failure of wind turbine subsystems from two wind farms in China showed that 68% total downtime was caused by generator, converter, and pitch systems [18].

Usually, SCADA systems provide data representing normal operation and faulty conditions. In some cases, we may not have enough data representing a healthy and faulty operation, perhaps due to broken sensors. In such a case, we can build a mathematical model of the equipment and estimate its parameters from sensor data. We can then simulate this model with different fault states under different operating conditions to generate failure data. We can then use the generated data to supplement our sensor data and use both to develop our algorithm. After completing the data acquisition, the next step is to remove the outliers and clean them up by filtering out the noise [9]. In this research, only sensor data representing the normal operating condition is available; we do not have data of faulty operation. We can build a predictive maintenance algorithm, but we would have to build a mathematical model of the wind turbine to generate failure data. This would require extensive domain knowledge of the system performance of the wind turbine. The following section will present a review of the different approaches that have used WT SCADA data for WT fault detection and prediction. In this paper, our contribution to knowledge will involve:

1) applying a purely data-driven approach to predictive maintenance using SCADA data without failure data
2) validate the process with a data from a different wind farm having failure data

This study will examine data on a wind farm (La Haute Borne) in France operated by ENGIE, where four 2MW wind turbine has been installed. Section II will describe related works on the ENGIE dataset and research that propose similar solutions in this paper. Section III developed the methodology to identify failures through data preprocessing, model development, and data post-processing. Section IV tests the developed method on a real case study of a wind farm currently operating in Meuse, France. Also, we evaluate our proposed solution against data from a wind farm with failure data. Section V discusses the effectiveness and applicability of the fault detection algorithm. Finally, section VI will be considering future steps.

II. LITERATURE REVIEW

In the last decade, predictive maintenance has been achieved by machine learning techniques used to build inductive models that learn the underlying set of structures in SCADA data of wind turbines to predict incipient faults and anomalies [10]. For the most part, many existing works utilise supervised methods, which can either be regression or classification; these methods have the advantage of providing a clear relationship between inputs and outputs [19]. This section will examine existing works based on regression-based anomaly detection and research on the ENGIE dataset.

A. REGRESSION-BASED ANOMALY DETECTION

This approach is used for condition monitoring in wind farms by building a model of the normal behaviour of the wind turbine and its components. A set of independent input(s) variables, such as wind speed, is used to build a regression model to predict a numeric dependent output variable such as power, assuming that the component is ideal. For example, power curve modelling of a wind turbine is a critical task since the power curves of WTs made available from manufacturers were explicitly tested to the location where turbines are located. This implies that the turbines were subject to a particular weather condition which is most likely different

from that of the installation site [20]. To solve this challenge, study [21] compared four data-mining approaches: cluster centre fuzzy logic, neural network, K-Nearest Neighbour and Adaptive Neuro-Fuzzy Inference System (ANFIS) to monitor wind turbine power output and detect deviations. Initially only one input variable wind speed and output variable power was used, but by adding wind direction and ambient temperature as inputs variables, the models had a better fit with the data. In this research, ANFIS - a machine learning algorithm which combines neural network with fuzzy theory - achieved the best performance. Modelling turbine components such a generator using machine learning was investigated by the study [22], here extreme gradient boosting (XGBoost) and long-short term memory (LSTM) were compared based on their mean absolute error (MAE). In this study, XGBoost outperformed LSTM in terms of MAE, and it was more computationally efficient, executing at 150 times faster than LSTM. The predicted results were then compared with field measurements to detect if an anomaly was present. The study [23] developed a framework for anomaly detection and parameter identifications; the LSTM network was incorporated into the neuronal structure of the auto-encoder neural network. Adaptive threshold based on support vector regression(SVR) was used to reduce false alarm rate for anomaly detection. The effectiveness of the proposed method was verified by a case study using SCADA data from a wind farm near the coast of the south of Ireland. The study [13] utilized the generator temperature and gearbox oil temperature in SCADA data to establish a normal temperature model of the wind turbine components. The residual between the predicted and actual value was calculated, and the trend was monitored using an exponentially weighted moving average (EWMA) control chart. The study also proposed a fixed threshold and dynamic threshold based on adaptive algorithm compared - their fault detection efficiency. The study [24] performed feature selection using an adaptive elastic network, and convolutional neural network (CNN) and LSTM were combined to establish a logical relationship between observed variables. The method was efficient to detect over-temperature in the high-speed side of the gearbox bearing. The research [25] proposed a model that detects abnormal spikes in wind turbine components by adjusting temperature data for effects caused by ambient temperature and when the turbine is outputting power. Regression models with inputs variables (power output and ambient temperature) and output variable (component temperature) were built. The best model, which in this case was linear regression, was selected. The residual between the model's output temperature and raw temperature data was used to detect abnormal behavior of the component. The study [26] carried out predictive analytics of wind turbine gearbox based on SVR models for accurate prediction of gearbox oil and bearing temperature. Diebold-Mariano and Durbin-Watson statistical tests were used to analyse the residuals to establish the robustness of the tested SVR model. The study [27] applied the Mahalanobis distance method for feature selection, which helped to reduce the input variables fed into the LSTM prediction model. The fault detection was carried out using the error between predicted component temperature and actual measurement. This method yielded more efficient and accurate results lowering root mean square error by 4% compared to the traditional backpropagation neural networks. The study [28] investigated the use of electrical parameters of SCADA measurements to build data-driven normal behaviour models constructed through SVR with Gaussian kernel to capture the non-linear relationship between the electrical parameters and operational variables. Principal components analysis (PCA) was used to orthogonalize and reduce features dimensions. The normal behaviour model of the healthy wind turbine and the target faulty wind turbine were analyzed in parallel; it was shown that the fault could be detected two weeks before it occurred. The study [29] proposed a comprehensive methodology for designing and applying artificial neural networks and statistical process control for effective fault detection of wind turbines. The proposed method was tested on an actual wind turbine in Italy to verify its effectiveness and applicability.

B. RELATED WORKS ON ENGIE WIND FARM DATASET

The study [19] proposed a novel idea of bringing together LSTM and XGBoost to predict an anomaly in wind turbines. The model was used on a source domain for learning on a labelled dataset (LDT dataset). The learning was transferred to the unlabelled dataset as the target domain (Engie dataset). The objective of the transfer learning was to enable wind farm operators with no access to historical data of failures to detect anomalies. The study [30] has developed a system for reconstructing the lost signal from low correlated parameters when one of the SCADA sensors fails to send data. The objective of the signal reconstruction model was for wind power prediction from other SCADA parameters. Linear and non-linear algorithms were analysed to find a generalised model, multiple linear regression random forest and, Cartesian genetic programming evolved Artificial Neural Network (CGPANN) was used to inform the generalised model. The study [23] proposed solution to high-dimensionality problems of condition monitoring (CM) data coming off mechanical equipment. Since this equipment presents multiple operating conditions, it is difficult to isolate the anomalies without mixing them up with the normal operating conditions of the equipment. Therefore, the Gaussian mixed model was employed to cluster the operating conditions. The isolation forest method was used to detect anomaly instances and identify the critical attributes responsible for the equipment degradation. This model was demonstrated on the ENGIE dataset to evaluate its effectiveness. The study [31] applied the novel improved dragonfly algorithm (IDA) to choose optimal parameters of support vector machine (SVM) for the forecast of short-term wind power. This hybrid model (IDA-SVM) outperformed the traditional grid search algorithm (Grid-SVM), which only compares different parameter combinations to select the best performance. In IDA-SVM, adaptive learning factors and differential evolution strategies were taken to boost the

optimisation ability of the dragon algorithm (DA), which was applied to the ENGIE dataset at different seasons. The study [32] used the ENGIE dataset as a validation set to show that the novel k-means-based Smoothing Spline hybrid model achieves the most accurate power curve in terms of better goodness of fit statistics. This is in comparison to other k-medoids++ -based Gaussian hybrid models.

III. METHODOLOGY

This study aims to investigate a robust and precise workflow for fault detection in wind turbines based on xgboost, LSTM, and Statistical Process Control (SPC). The methodology will outline steps to build a predictive maintenance system based on fault detection when we do not know what failure looks like, that is, the absence of failure data. However, there has been much study about predictive maintenance based on SCADA data using machine learning and SPC, as elaborated in section II. One common thing about the works is that they validated their solutions using the available failure data, maintenance logs, alarm logs, or status logs recorded in the wind farm. This study will validate our model's predictive ability based on data from a different wind farm having failure data by way of transfer learning. We will examine the effectiveness of our method to predict failure when there is no historical data on the maintenance of the wind farm.

The critical steps of our method are highlighted below:

1) Data Acquisition and data preprocessing: data is collected from open-source platforms, data cleaning, outlier removal, and filtering normal operational data points for subsequent model processing.
2) Model processing: the building of models for the turbines in the wind farm to represent normal behaviour.
3) Post-processing: the deviations of model predictions against actual measured data is evaluated using the SPC control chart.

We will build a model representative of the normal behaviour of the wind turbines with the assumption that our model will always provide information about the healthy state of the turbine. Next, we will predict the wind turbine's health status in the testing phase, this healthy representative state of the wind turbine will serve as a reference for asset managers. Therefore, when new SCADA data has been acquired, the deviations between the healthy wind turbine model are compared with the latest data. These deviations will be monitored through the SPC control chart; data points outside the allowable fault threshold are considered an anomaly. To validate this method, we will train our model on new data from a different wind turbine having failure data; only after this, the model is deemed ready for real-time monitoring. Fig.2 represents the fault detection algorithm based on temperature prediction of wind turbine components.

A. DATA ACQUISITION AND DATA PREPROCESSING

To build a healthy representative model, historical monitoring data of wind turbines spanning over a considerable period was obtained from a wind farm. Since SCADA data provide

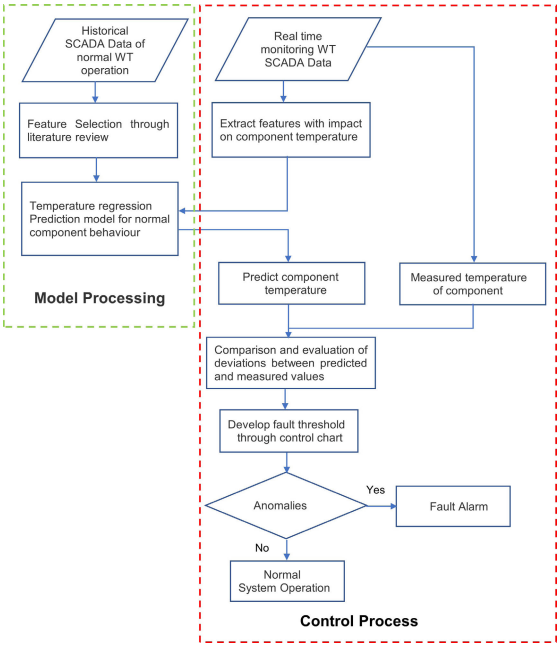


FIGURE 2. Block diagram of WT predictive maintenance fault detection algorithm.

helpful monitoring and control information in real-time, the data used in the study is SCADA data acquired from the La Haute Borne wind farm located in Meuse, France [33]. This wind farm is operated by ENGIE Green, having four wind turbines manufactured by Senvion MM82 technology. The SCADA system in this wind farm acquired data of 34 measured parameters as well as their statistics such as average, maximum, minimum, and standard deviation of each parameter. We only retain the average values of each parameter since it captures most of the information. The frequency of captured data points is sampled at the 10-min interval. The rated power for each turbine at the La Haute Borne wind farm is 2050kW, having a rotor diameter of 82m and a hub height of 80m. The cut-in wind speed is 3.5m/s, rated wind speed of 14.5m/s, and cut-out wind speed of 25m/s. The key parameters that we will consider in this study include active power, wind speed, outdoor temperature(ambient), generator bearing temperature, gearbox bearing temperature, Generator speed, Gearbox oil sump temperature, Rotor speed and Nacelle temperature.

1) DATA CLEANING

The algorithms used to train our models will build a relationship between the inputs and output variables. Therefore the data quality must be examined to ensure the model represents the system condition with the feed data. Any anomalous data points must be removed to avoid giving the model a wrong impression of system performance. To build a model representing the healthy state of the WT, data cleaning operations must be carried out. After identifying the variables needed for model processing, an understanding of system performance and the variables describing them in the data

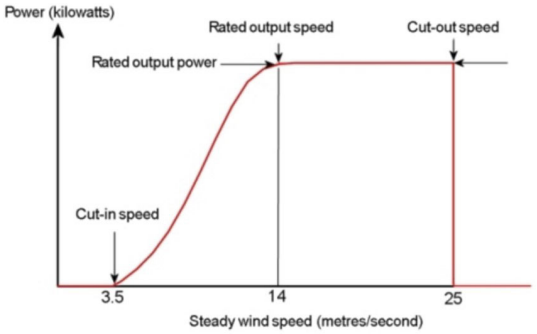


FIGURE 3. Typical Power curve for a wind turbine.

must be carried out. This enables us to identify anomalies in the data and remove them since they have a significant impact on our model accuracy. From the WT system performance, sensors are used to gather data from the SCADA system; therefore, there could be data spikes or no data due to sensor errors. This sensors errors can arise due to non-calibration of sensors or sensor degradation over time, creating outliers in SCADA data [34]. In addition to sensor malfunction, wind farms are subject to power reductions imposed artificially either due to maintenance or by the national grid to combat dispatching issues [29]. Therefore, the following elimination criteria were used for preliminary data cleaning:

- Instances where turbine power is zero or less, but wind speed is above cut-in speed
- Samples where at least one input or output is missing
- Samples with one or more values that are outside the normal range
- Samples where wind turbine was on halt or data loss because of sensor transmission errors

The summary of the data cleaning and resampling operation is shown in (1) [39].

$$\begin{cases} \text{delete } x_i, & \text{for } x_i \in \text{ hault data} \\ x_i = x_{i+1} \text{ or } x_{i-1}, & \text{for } x_i \in \text{ packet loss data} \\ x_i = 0, & \text{for } x_i < 0 \text{ or } x_i \text{ is null} \\ \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, & \text{others} \end{cases} \tag{1}$$

The SCADA data records power limit values, and this operation does not represent the turbine's ideal behaviour. The data points collected during such power restrictions must be removed from the dataset. After getting rid of abnormal data points, the second pass of cleaning must be done on the data to catch outliers due to unknown reasons.

2) POWER CURVE FILTERING

The power curve is used as a reference for the expected behaviour of the wind turbine, as seen in Fig.3. Hence data representing healthy is required to follow the power curve signature. The wind turbine power curve shows the relationship between wind turbine power and wind speed. It essentially captures the wind turbine performance. Hence it plays a vital role in condition monitoring and control of wind turbines.

TABLE 1. Input and output variables used for modelling different components.

Component	Inputs	Output	Ref.
Gearbox	Nacelle temperature Rotor speed Active power Outdoor temperature Gearbox oil sump temperature	Gearbox Bearing Temperature	[31] [33] [38]
Generator	Nacelle temperature Active power Generator speed Generator stator temperature	Generator Bearing Temperature	[11] [20] [35]

Power curves are made available by the manufacturers to help estimate the wind energy potential in a candidate site. The characteristics curve of a wind turbine behaves differently in different regions due to wind speed's intermittent and stochastic nature. Therefore, applying the traditional outlier detection methods usually fail to catch them or catches along with healthy data points. We are interested in fitting a power curve to data representing 'normal' turbine operation. In other words, we want to flag all anomalous data or data representative of underperformance. The study [29] recommends dividing(binning) the data into intervals where the turbine changes behaviour. After binning the samples, to detect outliers, we calculate the quantiles of the data within each bin and eliminate the outliers of the corresponding boxplot. The criterion for flagging is based on some measure (scalar or standard deviation) from the mean of the bin centre. A scalar measure was applied to determine the outliers consisting of the threshold value of 25% from the mean of the bin centre of the whisker length.

B. MODEL PROCESSING

1) FEATURE SELECTION

To describe the healthy behaviour of the wind turbine, the variables that will form the input and output must be known.

But it is difficult to know beforehand these variables since there are many parameters measured by the sensors that make up the SCADA system. In this study phase, we relied on the literature review to understand the best variable combinations needed to monitor the system behaviour of critical components like the gearbox and generator. The bibliographic search of the component variables covered various methods researchers have used to arrive at a list of the most influential variables. In Table.1, the input and output variables that define the behaviour of the components of interest, based on the scientific literature review, are displayed.

2) REGRESSION-BASED MODELS

Due to the stochastic nature of wind, the algorithm required to model wind turbines should adequately and accurately

capture the complex relationship between variables defining the system performance. We will examine the effectiveness of regression models to build characteristic healthy behaviour of WT components using the input variables and output variable. The dataset instances will be divided into training and testing with percentages of 70:30 for each component model. The model accuracy on the training set was compared with that of the test set to check for model overfitting. We also employed K-fold cross-validation five times to ensure the model is robust and accurate, preventing data leakage, overfitting, or underfitting. Because values of the input variables are in different dimensions and ranges, it is necessary to force their values within a given defined range. In this study, the input variables were standardised using the sklearn standard scalar function. The function essentially computes the z-score with mean and standard deviations of the variables and scales them to the interval [0,1]. The computation for transforming the selected input data to z-score is shown in (2).

$$Z = \frac{x_i - \mu}{\sigma} \tag{2}$$

where x_i is the set of input variables, μ is the mean, and σ is the standard deviation. The model will use the input variables of the training set to predict the output variables also belonging to this set, studying their underlying relationships. How well the model predicts the output variable is used to define the training accuracy. After that, the input variables belonging to the test set unseen by the model are used to predict the output variable. The accuracy of the model is then determined by how well the model can predict the output variable. We then will compare the predicted output variables representing the healthy condition of the WT to the measured values. We will start with a naïve model using multiple linear regression (MLR) as a baseline model, then compare it with two non-linear algorithms such as extreme gradient boosting (XGBoost) and long short-term memory (LSTM). Our algorithm choice is determined through the study of technical and scientific literature [22], [23].

a: MULTIPLE LINEAR REGRESSION (MLR) MODEL

Multiple linear regression models the relationship between two or more input variables and an output variable by fitting a linear equation to observed data. Every value of the input variable x is associated with a value of the output variable Y. It is a statistical technique used to predict the output variable Y from a set of input variables x_i where i is the index of the predictor variables as shown in (3).

$$Y = \beta_o + \beta_1 x_1 + \cdots + \beta_i x_i + \epsilon \tag{3}$$

The model parameter β_o which is the intercept of the fitted regression line, the regression coefficients (β_1, β_i) are learned during model training of the data, and ϵ is the model's deviation in Y. The transformed dataset is fed into Python's Scikit-learn linear regression algorithm.

b: XGBoost REGRESSION MODEL AND HYPERPARAMETER OPTIMIZATION

XGBoost is a machine learning algorithm that Dr Chen proposed in 2016 [35]. It is an ensemble model based on decision trees that combine multiple weak learners into strong learners through multiple iterative learning processes. It works by boosting numerous weak learners such as regression trees by assembling them to create a single but stronger learner [36]. The basic principle behind the process is to learn at each iteration sequentially, and the present regression tree is fitted with the residual from the previous three. In other words, the base learners' (weaker regression trees) mistakes or errors are learned and are used to correct the new regression tree. The new regression tree is added to the fitted model to update the residuals while an objective function tracks the models' performance changes. The objective function has a regularisation term that penalises the model complexity to prevent overfitting of the model output and helps better generalise the model's ability. XGBoost uses the loss function of the base models to minimise the residual of the overall model. To do this efficiently, XGBoost uses first and second-order partial derivative estimations to gain information about the direction of gradients [22]. The XGBoost exhibits faster model exploration by using all the CPU cores in a parallel and distributed manner during the training process, which helps it to reduce the training computation time and complexity and ensures faster learning [37].

XGBoost Regression Model: Since the entire process is an ensemble model of CART (classification and Regression Tree) having decision tree as the based model, the output of model $\hat{y}_{Xi}$ is voted or averaged by a collection F of m trees shown in (4).

$$\hat{y}_{Xi} = \sum_{m=1}^{M} f_m(x_i), f_m \in \tag{4}$$

where $\hat{y}_{Xi}$ denotes the predicted value of the i-th sample, M denotes the number of CART in the model, $f_m(x_i)$ represents the predicted value of the i-th sample in the m-th tree, F is the function space of CART. The objective function of the XGBoost includes the MSE loss function and the regularisation term represented by (5) [36].

$$\begin{cases} Obj = \sum_{i=1}^{\eta} l\left(y_{Xi}, \hat{y}_{Xi}\right) + \sum_{m=1}^{M} \Omega\left(f_m\right) \\ \Omega\left(f_m\right) = \gamma T + \frac{1}{2}\beta \sum_{j=1}^{T} w_j^2 \end{cases} \tag{5}$$

where η denotes the number of samples, l denotes a second-order derivable loss function, which measures the difference between the actual value y_{Xi} and the predicted value $\hat{y}_{Xi}$.$\Omega(f_m)$ represents the regularization term. T is the number of leaf nodes in the tree, w_j is the score of the leaf nodes, γ and β are the parameters to control the complexity of the tree. The purpose of optimising the objective function is to determine the structure of CART, that is, to get the

best-split feature and the best split point and the leaf node score w_i. The objective function can be simplified to a unitary quadratic equation as a function of w_j by the second-order Taylor expansion represented in (6). More details on the expansion simplification can be found in [14].

$$\text{Obj} = \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right) w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \beta\right) w_j^2\right] + \gamma T \tag{6}$$

where I_j represents all the data samples in the leaf node j, g_i and h_i denote the first and second derivatives of the MSE loss function. From (6), G_j and H_j can be defined as in (7) [35]:

$$G_j = \sum_{i\in I_j} g_i, H_j = \sum_{i\in I_j} h_i \tag{7}$$

The optimal score of the leaf node w^*, represented by (8). And the corresponding optimal value of the objective function *Obj* represented by (9) is obtained by solving the unitary quadratic (6) with the assumption that the structure of the CART is known.

$$w_j^* = -\frac{G_j}{H_j + \beta} \tag{8}$$

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \beta} + \gamma T \tag{9}$$

A smaller value of the objective function provides a better structure of the CART. XGBoost applies a greedy algorithm to navigate all the split points and finally selects the split point with the smallest value of the objective function after splitting. This means that the optimal split point is chosen at the maximum gain as represented in (10).

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon l_l} g_i\right)^2}{\sum_{i\in I_L} h_i + \beta} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \beta} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \beta}\right] - \gamma \tag{10}$$

where I_L and I_R are the data sample sets of left and right nodes after splitting, I denotes the union sets of I_L and I_R.

XGBoost Hyper-Parameters Optimization: Typically, machine learning models' performance gets better on tuning their hyper-parameters. For XGBoost, there are more than ten hyper-parameters that require manual setting of their values to build a regression model. The hyper-parameters have three categories: general parameters, task parameters, and booster parameters. By design and through experimental results, the boosting parameters possess the most significant impacts on the model's performance. To buttress this point, a critical look at one of the boosting parameters, eta, is used to update the weight of the leaf nodes. To keep the gradient in check as well as prevent it from being too big, the score of the leaf node is multiplied by the eta in each iteration. If the model has a small value of eta, then it is more likely to overfit, but if the eta value is too large, the model is expected to underfit. It is now clear how significantly the choice XGBoost hyper-parameters improves its performance [36]. Determining the best hyper-parameters can be a painful task if one is required to perform them manually. Hence three effective techniques are used to select the best combinations of hyper-parameters algorithmically. These are random search optimisation [39],grid search optimization and Bayesian optimization algorithm [40]. This study employs the random search optimisation technique to iterate over the dynamic model to obtain the hyper-parameter best combination that optimises the model. Because this technique scales faster for large datasets than grid search, this method fit our study. The random search optimisation algorithm works by setting up a grid of hyper-parameter values and selecting the combinations that train and evaluate the model [21]. The tuning space of the hyper-parameters is shown in Table.2 And the optimal parameters are shown in Table.3

TABLE 2. XGBoost hyper-parameter tuning.

Hyper-parameters	Random Search Tuning Space
Maximum tree depth for base learners	6, 10, 15, 20
Boosting learning rate ("eta")	0.001, 0.01, 0.1, 0.2, 0.3
Subsample ratio of the training instance	0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Subsample ratio of columns when constructing each tree	0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Subsample ratio of columns for each level	0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Minimum sum of instance weight(hessian) needed in a child	0.5, 1.0, 3.0, 5.0, 7.0, 10.0
Minimum loss reduction required to make a further partition on a leaf node of the tree	0, 0.25, 0.5, 1.0
L2 regularization term on weights (xgb's lambda)	0.1, 1.0, 5.0, 10.0, 50.0, 100.0
Number of decision trees	30, 40, 50, 60, 70, 90, 80, 100
objective function	reg: squarederror

TABLE 3. XGBoost optimal hyper-parameters.

Hyper-parameters	Optimal parameters
Maximum tree depth for base learners	10
Boosting learning rate ("eta")	0.1
Subsample ratio of the training instance	0.7
Subsample ratio of columns when constructing each tree	0.8
Subsample ratio of columns for each level	0.9
Minimum sum of instance weight(hessian) needed in a child	1.0
Minimum loss reduction required to make a further partition on a leaf node of the tree	0.5
L2 regularization term on weights (xgb's lambda)	0.1
Number of decision trees	60
objective function	reg: squarederror

c: LONG SHORT-TERM MEMORY (LSTM)

The long short-term memory (LSTM) mimics the human's ability to interpret the meaning of a word from the context of the entire sentence. Similarly, LSTMs produce predictions

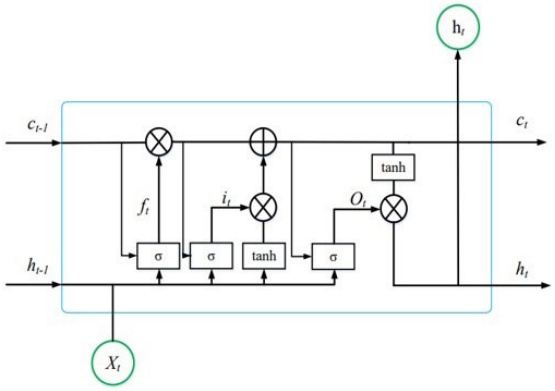


FIGURE 4. Single layer of an LSTM cell.

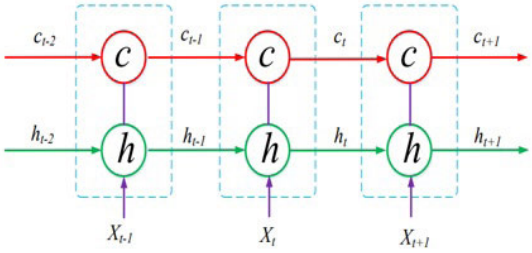


FIGURE 5. Temporal-logic framework of a single LSTM layer.

from an ordered sequence of temporal data they receive as inputs. A typical example of such data is SCADA logs which have successive time intervals. Hochreiter and Schmidhuber first proposed the LSTM in 1997 [41] as special type of recurrent neural network (RNN) to overcome the incipient vanishing and exploding gradients problems in RNN. The LSTM's ability to learn the long-term and short-term dependencies inherent in a sequential data has made it more successful in predicting long input sequences such as that found in SCADA data [18]. The architecture of the LSTM algorithm is shown in Fig.4 and Fig.5. This algorithm possesses feedback connections and can define non-linear dynamic systems by mapping input sequences to output sequences. The basic structure of this algorithm possesses a cell and three gates (input gate, output gate, and forget gate); the cell acts as the memory of each LSTM unit; the gates control information flow with each LSTM unit [26].

The solution to RNN's long-term dependencies problem lies in the cell state of the LSTM structure. This cell state's main purpose is to store long-term information in the LSTM's hidden layer. From Fig.5, X_t is the present-time input vector, which is the input data to the LSTM model at time t; $\mathrm{h_{t-1}}$ is the past time output vector; and $\mathrm{c_{t-1}}$ represents the past time cell state. In Fig.4 f_t and i_t is the forget gate and the input gate respectively, which are used to control the cell state of the model. In other words, forget gates and input gates are fashioned to restrict the information flow. σ is a sigmoid function deciding which values to be updated in the cell state and outputs a number between 0 and 1 for each number in the cell state c_{t-1}. Where 1 represents "completely keep this" while a 0 represents "completely get rid of this" [42]. The forget gate controls the past cell state information c_{t-1} transmitted to the present cell state. The process can be explained with (11),

$$\mathrm{f_t} = \mathrm{g}\left(\mathrm{W_f} \cdot [\mathrm{h_{t-1}}, \mathrm{X_t}] + \mathrm{b_f}\right) \tag{11}$$

where $g(\cdot)$ is the activation function that executes the sigmoid nonlinear function, W_f represent the forget gate weight matrix, b_f is the bias vector of the forget gate, and $[h_{t-1}, X_t]$ is the combination vector of the past time output vector h_{t-1} and the present time input vector X_t. The input gate $\mathrm{i_t}$ controls the present input $\mathrm{X_t}$ information transmitted to the current cell state $\mathrm{c_t}$, shown by (12).

$$\mathrm{i_t} = \mathrm{g}\left(\mathrm{W_i} \cdot [\mathrm{h_{t-1}}, X_t] + \mathrm{b_i}\right) \tag{12}$$

where W_i is the weight matrix of the input gate, b_i is the bias vector of the input gate. To capture the state of the current input, $\mathrm{c_t'}$, can be calculated as seen in (13).

$$c_t' = \tanh\left(W_c \cdot [h_{t-1}, X_t] + b_c\right) \tag{13}$$

where W_c and b_c are the weight matrix and the bias vector, respectively, tanh is the hyperbolic tangent function (which distributes values of the cell state between -1 and 1). The present cell state c_t can then be obtained by combining both the forget gate and the input gate, described by (14).

$$c_t = f_t * c_{t-1} + i_t * c_t' \tag{14}$$

where * defines element-wise multiplication between vectors, the information flow from the present cell state c_t is controlled by the output gate O_t to the current output, described by (15).

$$\mathrm{O_t} = \mathrm{g}\left(\mathrm{W_o} \cdot [\mathrm{h_{t-1}}, \mathrm{X_t}] + \mathrm{b_o}\right) \tag{15}$$

where W_o is the weight matrix, b_o is the bias vector. Lastly, the output gate O_t and the current cell state c_t determine the output of LSTM model represented in (16).

$$\mathrm{h_t} = \mathrm{O_t} * \tanh\left(\mathrm{c_t}\right) \tag{16}$$

The complex nature of SCADA data with its non-linear multivariate time series makes LSTM a perfect candidate to capture the long-term dependencies inherent in them. Also, LSTM can eliminate the need for manual feature engineering by identifying optimal features automatically [19]. The model hyperparameters and configuration is shown in Table.4 And Table.5.

3) EVALUATION METRICS OF MODELS PREDICTIVE ABILITY

In this study, four metrics was used to effectively evaluate the temperature predictive regression models discussed in section II above. These metrics are viz: coefficient of determination(R-Squared), root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). The corresponding (17) to (20) shows the calculation formulas for the relevant metrics:

$$R^2 = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2 / \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \bar{y}_i\right)^2 \tag{17}$$

TABLE 4. LSTM hyper-parameters.

Hyper-parameters	Used Values
Number of hidden LSTM layers	4
Activation function of output layer	ReLU
Loss function	MSE
Optimizer	ADAM
Batch size	5
Epochs	10

TABLE 5. LSTM hidden layer configuration.

Hidden Layers	Number of Neurons
LSTM Layer 1	50
LSTM Layer 2	50
LSTM Layer 3	25
LSTM Layer 4	1

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2} \tag{18}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|\left(y_i - \hat{y}_i\right)\right| \tag{19}$$

$$MAPE = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{\left(y_i - \hat{y}_i\right)}{\left(\hat{y}_i\right)}\right| \times 100\% \tag{20}$$

where y_i is the measured value, $\hat{y}_i$ is the predicted value, and m number of instances of data in the test set and $\bar{y}_i$ is the mean of the measured value. R^2 is also known as the goodness of fit. This defines the degree to which the regression model fits the observed values. The closer the value is to 1, the better the degree of the fitting and vice versa. Whereas for RMSE, MAE and MAPE, the smaller their values, the higher the accuracy of the prediction model. RMSE is very sensitive to errors, MAE is more robust to outliers; whereas MAPE cannot handle extremely small observed values close to zero or zero, but RMSE and MAE can handle such. Therefore, these strengths and weaknesses of the different metrics inform our choice of using the RMSE, MAE and, MAPE, in addition, R^2 to complement each other.

C. POST-PROCESSING

After training, testing, and evaluating the model's accuracy, we proceed to assess and compare the deviations calculated as in (21).

$$\Delta = \text{measured values} - \text{predicted values} \tag{21}$$

To evaluate the deviations between each instance of sensor reading (measured sensor value and predicted values by the model), we used statistical process control to identify anomalies in the WT. For this study, the Shewhart control chart is used to evaluate the deviations as it evolves. The fault threshold is defined by two control limits used to evaluate abnormal behaviours: Upper Control Limit (UCL) Lower Control Limit (LCL) The two control limits describe the sensitivity of the control chart, which is expressed as multiples of σ of the deviation's distribution. Where σ is standard deviation calculated from the moving range MR, which is the difference between the i-th deviation and the last one, (22) to (25) shows the process of computing the control chart sensitivity:

$$MR = |\Delta_i - \Delta_{i-1}| \tag{22}$$

$$\sigma = \frac{\overline{MR}}{1.128} \tag{23}$$

$$UCL = +3\sigma \tag{24}$$

$$LCL = -3\sigma \tag{25}$$

The predicted values depict the healthy state of the WT over time. The WT whose measured value is compliant with the healthy state will have deviations on the chart with the normal distribution. With a mean of zero and standard deviation of 1, whereas the presence of non-conformity is exhibited by randomness. Non-conformity is defined by data points beyond the fault threshold/control limits, shifts of the average. Hence, such signals are considered abnormal behaviour. The model is used to reveal incidents faults in a WT without failure data and validated using a different WT with maintenance logs to reveal real faults, and then the model is fit for purpose.

IV. RESULTS

In this section, we will present the application of the proposed methodology on two wind farms: La Haute Borne wind farm operated by ENGIE in Meuse, France, and a wind farm operated by EDP (Energias de Portugal) in the West African Gulf of Guinea [43], [44]. La Haute Borne dataset does not have any failure data or maintenance logs, whereas the EDP dataset has failure data in addition to operational data. In this study, we illustrate the usefulness of our prediction model if we do not have failure data in a wind farm. Especially when the WT is newly installed and running for a short period, we can harness that short period of operational data to predict when the WT will fail using the algorithm in this paper. First, we demonstrate this on the ENGIE dataset. After that, we validate our proposed methodology on EDP data which has maintenance data to show the algorithm's effectiveness to detect the fault.

A. ENGIE DATASET

1) DATA PREPROCESSING

The data available is SCADA data recorded every 10 min from January 1, 2013, to December 31, 2016, for a total of 136 sample variables for four turbines. Thirty-four unique parameters were recorded, and their basic statistics such as minimum, maximum, mean, and standard deviation.

We deleted variables with min, max, and std since the average values captured the most information. Not a number(Nan)

values handling were based on a set threshold. Variables with more than 10k Nan values were removed. This is because filling these values with any strategy can give a false impression of the WT conditions. Then the stepwise method in the data cleaning subsection of section III was followed to exclude data of the following categories:

- Instances where active power is zero
- Instances With at least one variable (input or output variables) of interest is missing
- Instances where WT is operating under the restricted power regime

Power restrictions are typically constraints put on wind farm operators by the national grid to prevent dispatch problems. These behaviours do not characterise normal WT operation; this influences our choice to remove instances affected by such limitations. Fig.6(a) shows one of the WT; many outliers are present. About 23% of instances were eliminated in the primary cleaning phase, affected mainly by the power restriction regime. After applying the primary cleaning procedure heightened in the data cleaning subsection of section III, we obtain a cleaner version of the power curve in Fig.6(b). In Fig.6(c), the clean data is obtained by applying the process laid out in the power curve subsection of section III. After getting the clean data ready for model training, the models' input parameters are extracted from the cleaned SCADA data to construct the input data set. The input data set variables making up the model for each WT component were selected based on the Table1. In Fig.7 and Fig.8 we have the graphical display of the chosen output variables needed to define WT models of its components (i.e., gearbox and generators) across the four turbines in the wind farm in the training phase.

Since we do not have maintenance records in this case study, we assume the turbine is in normal condition throughout its operation; therefore, we select all data for each of the eight models to analyse its behaviour in the training phase. A set of variables required to build each WT component was chosen to form an input dataset and the corresponding output variable. The input data set was standardised as discussed in Section III; after that, we split the entire dataset of the input dataset and the output variable. The split was done by selecting the first 70% of the data for training and the last 30% for testing. This was done to prevent data shuffling since the dataset is composed of time series so that any random data selection could result in data leakage.

2) MODEL PROCESSING

To build a robust fault detection model, all the algorithms discussed in Section III were used. MLR, XGBoost, and LSTM algorithms were used for each of the eight models, and the best model was selected based on combination of the performance metrics discussed in Section III. As mentioned in section III, the hyperparameters for XGBoost and LSTM were used for the eight models. This resulted in the varied performance of the algorithms where LSTM outperformed XGBoost in some models and vice versa. Multiple linear regression (MLR) performance was the poorest in all eight models; this further confirms that the relationship between the variables is non-linear. All the models were developed in Python 3.8.

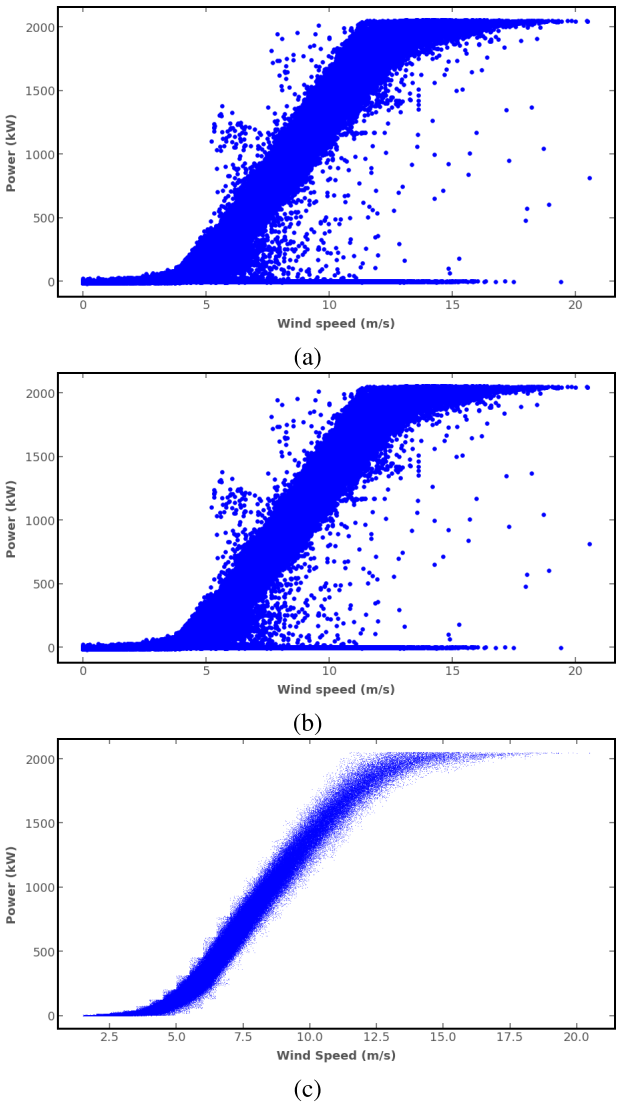


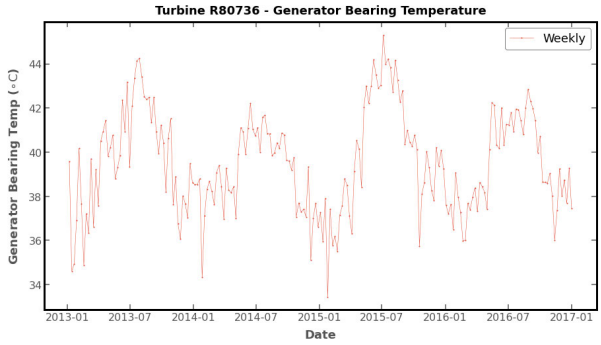
FIGURE 6. Wind turbine power curve: (a) Before data cleaning (b) after primary cleaning process (c) after the power curve filtering process.

a: GENERATOR MODEL

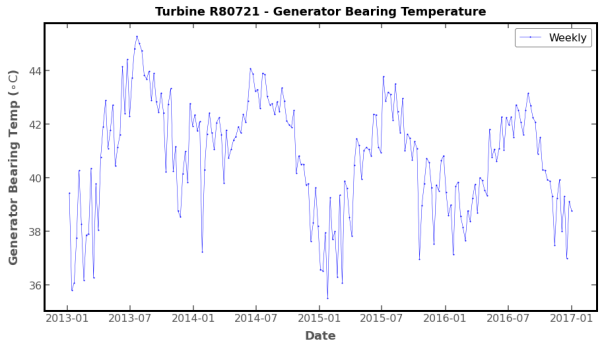
Wind Turbine R80736: After trying out the three models on WT R80736, LSTM model with RMSE = 1.44°C, MAE = 1.03°C, $MAPE$ = 2.65% was chosen because it had the highest $R-$ squared value and the lowest RMSE depicted having a lower MAE and MAPE than XGBoost as seen in Table.6.

This is because its R-Squared value shows better goodness of fit than other models; this means that the data fits the LSTM model better.

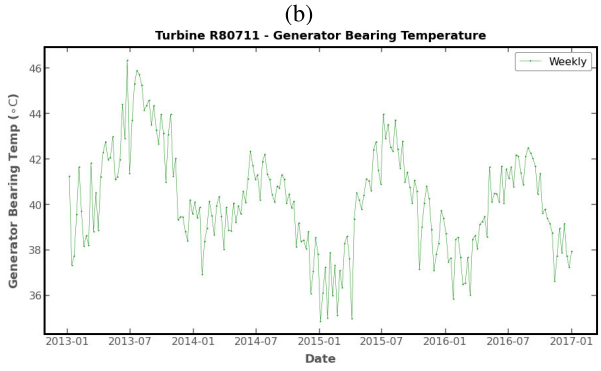
Additionally, because RMSE metrics penalise errors with higher values by assigning higher weights, the LSTM model is more accurate because significant errors are particularly



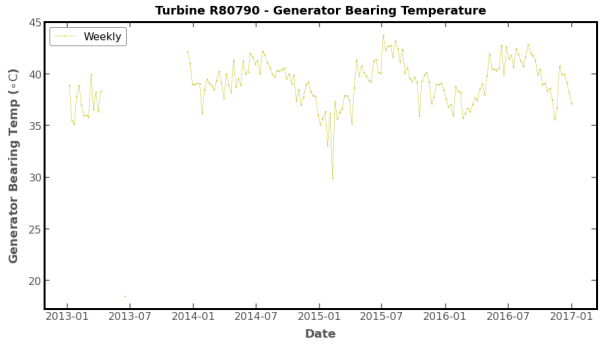
(a)



(b)

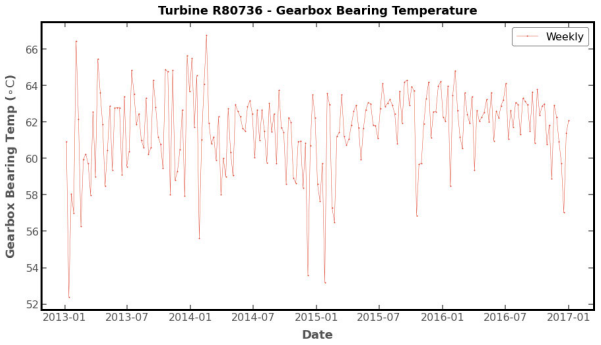


(c)

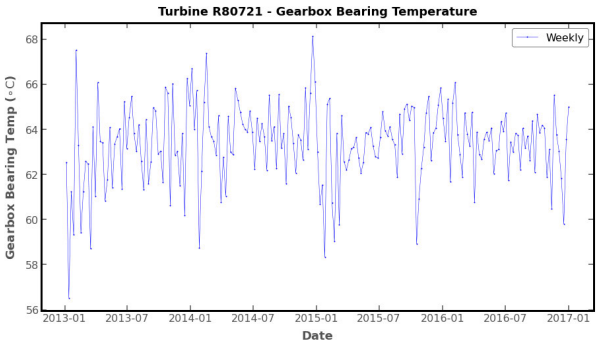


(d)

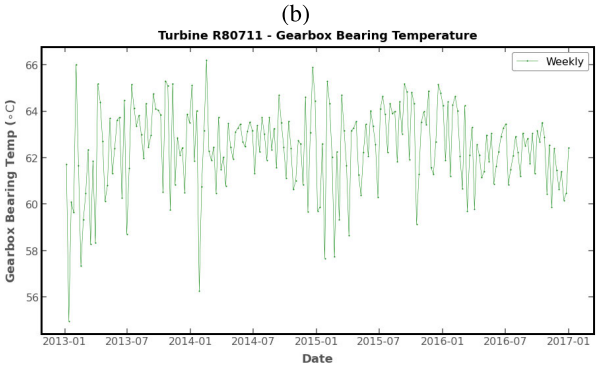
FIGURE 7. Graphical display of selected output variables during training phase WT generator for the four wind turbines in the wind farm. (a) Generator bearing temperature for Turbine R80736 (b) Generator bearing temperature for Turbine R80721 (c) Generator bearing temperature for Turbine R80711 (d) Generator bearing temperature for Turbine R80790.



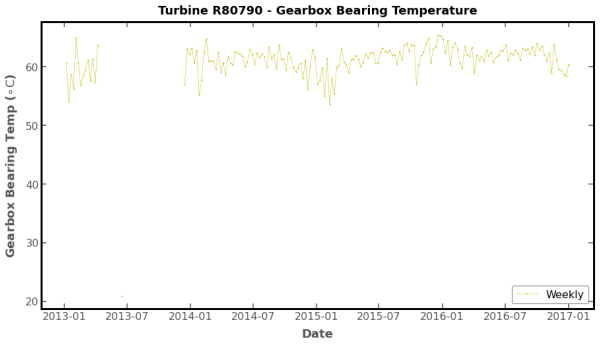
(a)



(b)



(c)



(d)

FIGURE 8. Graphical display of selected output variables during training phase WT gearbox for the four wind turbines in the wind farm. (a) Gearbox bearing temperature for Turbine R80736 (b) Gearbox bearing temperature for Turbine R80721 (c) Gearbox bearing temperature for Turbine R80711 (d) Gearbox bearing temperature for Turbine R80790.

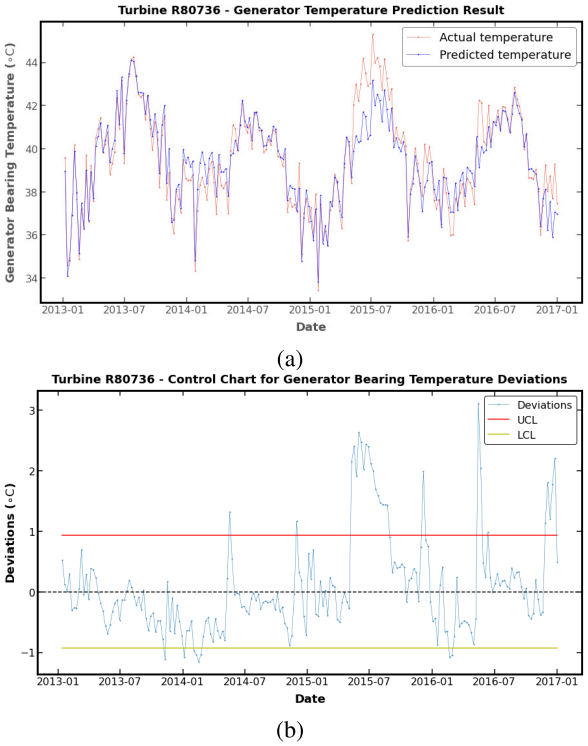


FIGURE 9. WT R80736 Generator: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

undesirable in our application. The LSTM model used the historical SCADA data from sensors recording the generator bearing temperature in WT R80736 during normal operation (healthy state) to predict the temperature, as shown in Fig.9(a). The control chart for this application is represented in Fig.9(b) with a fault threshold of ±0.93°C. The first point that wandered out of control was noted on November 10, 2013, but this was not significant as there was a slight shift in the average. However, on May 18, 2014, there was a significant point out of control, and on November 30, 2014, another point was out of control, showing a substantial shift in average. These events culminated in the spike of deviations from May 10, 2015, on the same side of the control chart, corresponding to the period where we have a massive spike in the actual generator bearing temperature in Fig.9(a). Therefore, this model predicted about four months about the imminent fault in the WT, which we assumed occurred around May 10, 2015, as shown by the evidence presented in Fig.9.

Wind Turbine R80721: For WT R80721, XGBoost model with $RMSE = 1.06°C$, $MAE = 0.8°C$, $MAPE = 2\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE and MAPE compared to LSTM and MLR as seen in Table.6. The XGBoost model used the historical SCADA data obtained by sensors recording the generator bearing temperature in WT R80721 during normal operation (healthy state) to predict the temperature as shown in Fig.10(a). The control chart for this application is represented in Fig.10(b) with a fault threshold of ±0.49°C. The first set of

TABLE 6. Model accuracy for generator bearing temperature prediction.

Turbine	Model	R-Squared	RMSE	MAE	MAPE
R80736	XGBoost	0.87664	1.46817	1.00013	0.0255
	LSTM	0.88042	1.44555	1.03447	0.0265
	MLR	0.85049	1.61632	1.18042	0.03048
R80721	XGBoost	0.93564	1.05955	0.80052	0.02048
	LSTM	0.92127	1.17181	0.9382	0.02381
	MLR	0.89748	1.33721	1.0517	0.02716
R80711	XGBoost	0.92555	1.0942	0.79672	0.02036
	LSTM	0.92827	1.07403	0.78204	0.01997
	MLR	0.87877	1.39622	1.09225	0.02835
R80790	XGBoost	0.81492	1.71331	1.1049	0.02744
	LSTM	0.81591	1.70871	1.05146	0.02603
	MLR	0.79268	1.81331	1.24832	0.03142

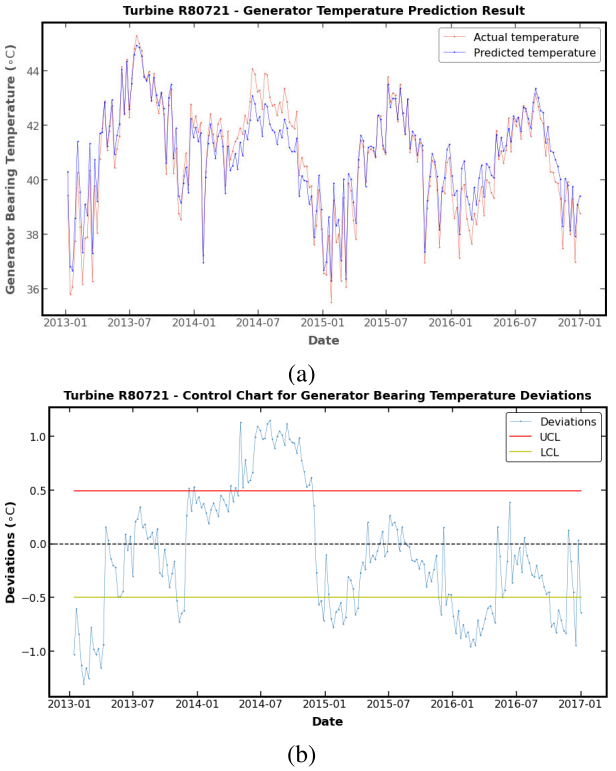


FIGURE 10. WT R80721 Generator:(a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

points below the fault threshold, beginning from January 13, 2013, to April 3, 2013, indicates that the WT must have been affected by the ambient temperature as is it winter in the wind farm location. A shift in the average started from December 8, 2013, and another point on December 22, 2013; this led to the significant deviation in magnitude on May 4, 2014. This period also synchronises with the increasing trend in generator bearings' actual temperature, as shown in Fig.10(a). We can say that the fault detection algorithm was able to predict the significant variations in WT performance that occurred on May 4, 2014, as early as December 8, 2013.

Wind Turbine R80711: For WT R80711, LSTM model with $RMSE = 1.07°C$, $MAE = 0.78°C$, $MAPE = 2\%$ was chosen because it had the highest R-Squared value; lowest

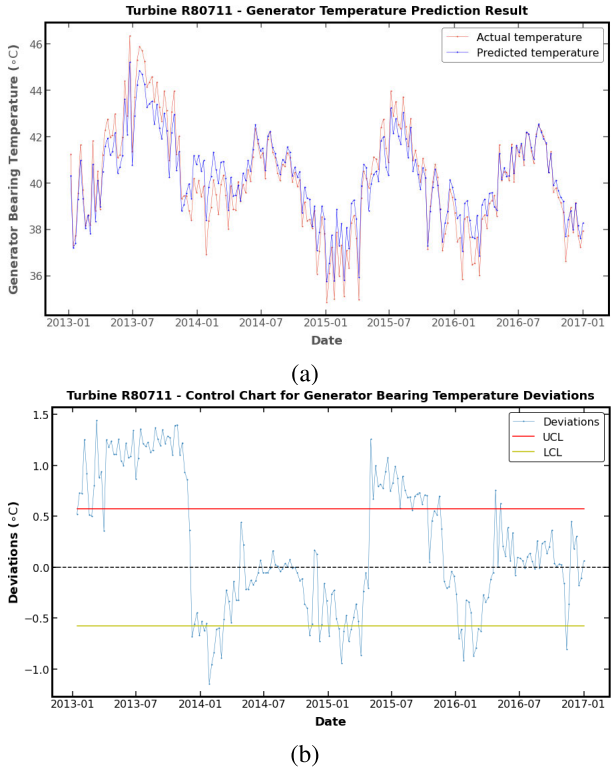


FIGURE 11. WT R80711 Generator: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

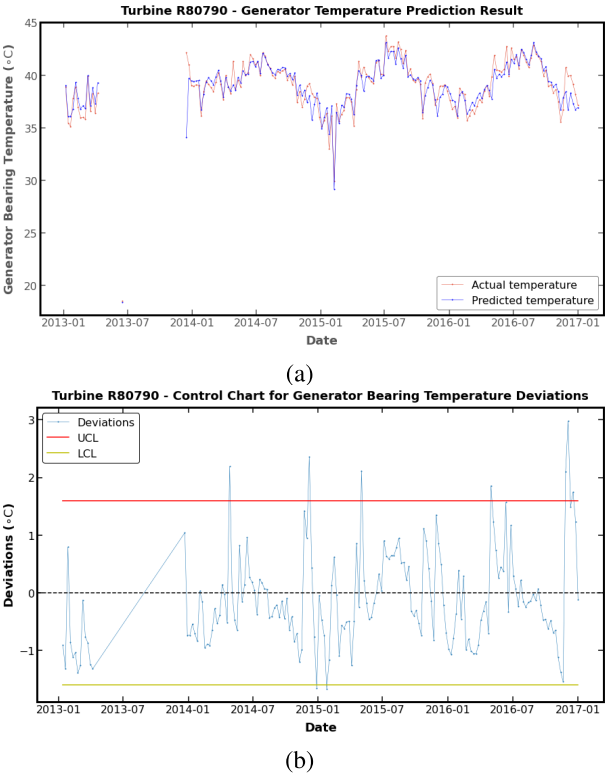


FIGURE 12. WT R80790 Generator: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

RMSE, MAE and MAPE compared to XGBoost and MLR as seen in Table.6. The LSTM model used the historical SCADA data obtained by sensors recording the generator bearing temperature in WT R80711 during normal operation (healthy state) to predict the temperature as shown in Fig.11(a). The control chart for this application is represented in Fig.11(b) with a fault threshold of $\pm 0.53°C$. Although there are numerous points out of control, there are still not enough elements to identify possible faults in the system, as seen in Fig.11(b). However, the general trend observed in Fig.11(a) is an upward trend of the actual and predicted temperature of the generator bearing.

Wind Turbine R80790: For WT R80790, LSTM model with $RMSE = 1.7°C$, $MAE = 1.05°C$, $MAPE = 2.6\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE and MAPE compared to XGBoost and MLR as seen in Table.6. The LSTM model used the historical SCADA data from sensors recording the generator bearing temperature in WT R80790 during normal operation (healthy state) to predict the temperature, as shown in Fig.12(a). The control chart for this application is represented in Fig.12(b) with a fault threshold of $\pm 1.46°C$. It is shown that the WT was out of service for a considerable amount of time from March 12, 2013, to December 28, 2013. The erratic temperature deviation was seen on December 12, 2016, almost two times more than the control limit and had an apparent out-of-point variation on May 1, 2016.

TABLE 7. Model accuracy for gearbox bearing temperature prediction.

Turbine	Model	R-Squared	RMSE	MAE	MAPE
R80736	XGBoost	0.96740	0.97504	0.70762	0.01182
	LSTM	0.95030	1.20396	0.95134	0.01575
	MLR	0.95363	1.16296	0.84495	0.01426
R80721	XGBoost	0.94872	1.12517	0.77478	0.01278
	LSTM	0.94098	1.20717	0.84654	0.01402
	MLR	0.92903	1.32377	0.90362	0.01507
R80711	XGBoost	0.95692	1.09161	0.80965	0.01359
	LSTM	0.95674	1.09382	0.80349	0.01364
	MLR	0.93844	1.30490	0.94409	0.01610
R80790	XGBoost	0.95596	1.08635	0.84781	0.01390
	LSTM	0.95402	1.11008	0.83334	0.01398
	MLR	0.93405	1.32938	1.01264	0.01688

b: GEARBOX MODEL

Wind Turbine R80736: For WT R80736, XGBoost model with $RMSE = 0.97°C$, $MAE = 0.71°C$, $MAPE = 1.2\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE and MAPE compared to LSTM and MLR as seen in Table.7.

The XGBoost model used the historical SCADA data obtained by sensors recording the gearbox bearing temperature in WT R80736 during normal operation (healthy state) to predict the temperature, as shown in Fig.13(a). The control chart for this application is represented n Fig.13(b) with a fault threshold of $\pm 0.25°C$. Although it is observed that some points are out of control at the beginning of the control chart from March 10, 2013, through to January 12, 2014, we do not have enough elements to validate if this event

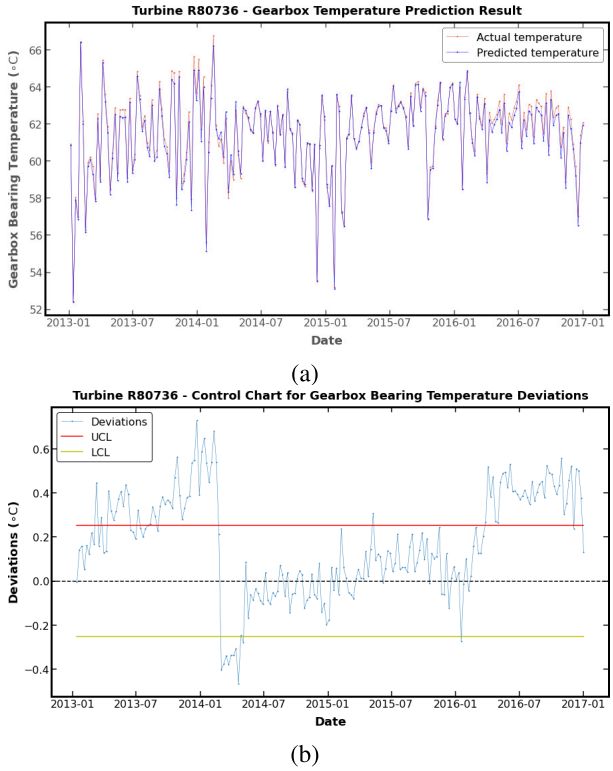


FIGURE 13. WT R80736 Gearbox: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

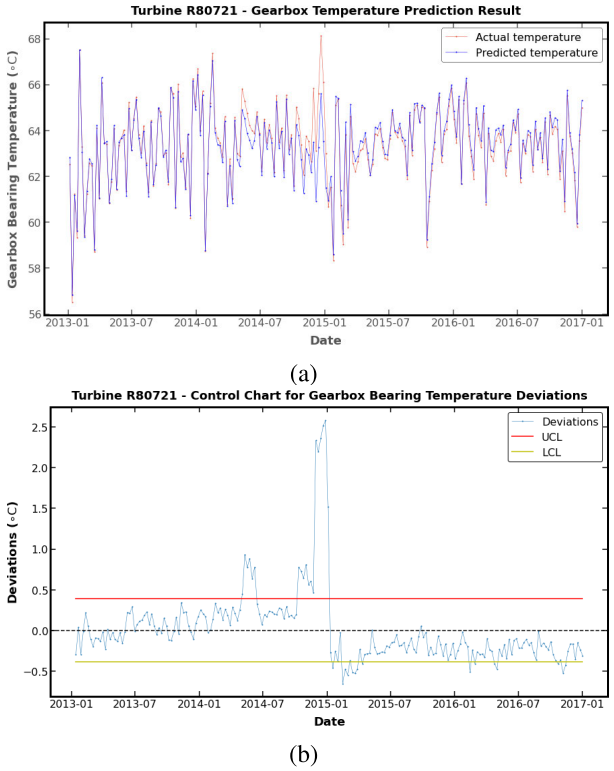


FIGURE 14. WT R80721 Gearbox: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

was fault-driven. But looking at the event towards the end of the control chart, we see numerous points outside the fault threshold starting from April 3, 2016, through to December 25, 2016. Before this extended dramatic event, we observed an out-of-control point on May 10, 2015. We can infer that this point warned of the possible fault events towards the end of the control chart.

Wind Turbine R80721: For WT R80721, XGBoost model with $RMSE = 1.12°C, MAE = 0.77°C, MAPE = 1.3\%$ was chosen because it had the highest R-Squared value, lowest RMSE, MAE, and MAPE compared to LSTM and MLR, as seen in Table.7. The XGBoost model used the historical SCADA data obtained by sensors recording the gearbox bearing temperature in WT R80721 during normal operation (healthy state) to predict the temperature, as shown in Fig.14(a). The control chart for this application is represented in Fig.14(b) with a fault threshold of $\pm 0.39°C$. The first set of out-of-control points that started from May 4, 2014, to June 8, 2014, culminated in a massive spike of about six times the fault threshold as seen in Fig.14(b) from November 30 2014 to December 28, 2014. This corresponds with the spike we see in Fig.14(a) of the actual temperature. We can say that our algorithm predicted the anomaly that occurred from November 30, 2014, to December 28, 2014, about five months ahead.

Wind Turbine R80711: For WT R80711, XGBoost model with $RMSE = 1.09°C, MAE = 0.81°C, MAPE = 1.3\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE and MAPE compared to LSTM and MLR as seen in Table.7. The XGBoost model used the historical SCADA data obtained by sensors recording the gearbox bearing temperature in WT R80711 during normal operation (healthy state) to predict the temperature as shown in Fig.15(a). The control chart for this application is represented in Fig.15(b) with a fault threshold of $\pm 0.34°C$. From Fig.15(b), we can see that from March 10, 2013, there has been a shift in the average of the fault threshold with an upward trend in the deviations, which led to the massive spike of the variation seen on November 24, 2013. Assuming there is planned maintenance annually in the wind farm that is carried out between January and December of the year, our algorithm was able to predict the fault before the annual intervention.

Wind Turbine R80790: For WT R80790, XGBoost model with $RMSE = 1.09°C, MAE = 0.85°C, MAPE = 1.4\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE, and MAPE compared to LSTM and MLR as seen in Table.7. The XGBoost model used the historical SCADA data obtained by sensors recording the gearbox bearing temperature in WT R80790 during normal operation (healthy state) to predict the temperature as shown in Fig.16(a). The control chart for this application is represented in Fig.16(b) with a fault threshold of $\pm 0.33°C$. It is shown that the WT was out of service for a considerable amount of time from March 12, 2013, to December 28, 2013. Although there has been a massive spike in the deviations from

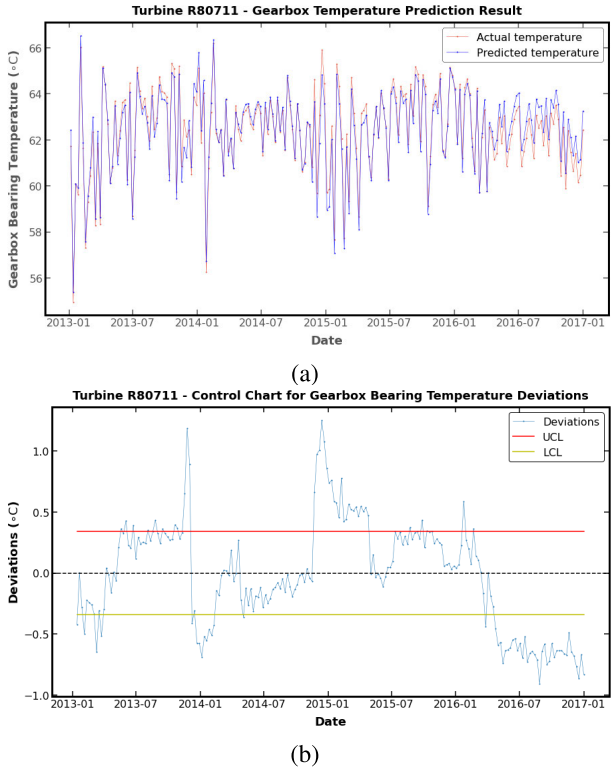


FIGURE 15. WT R80711 Gearbox: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

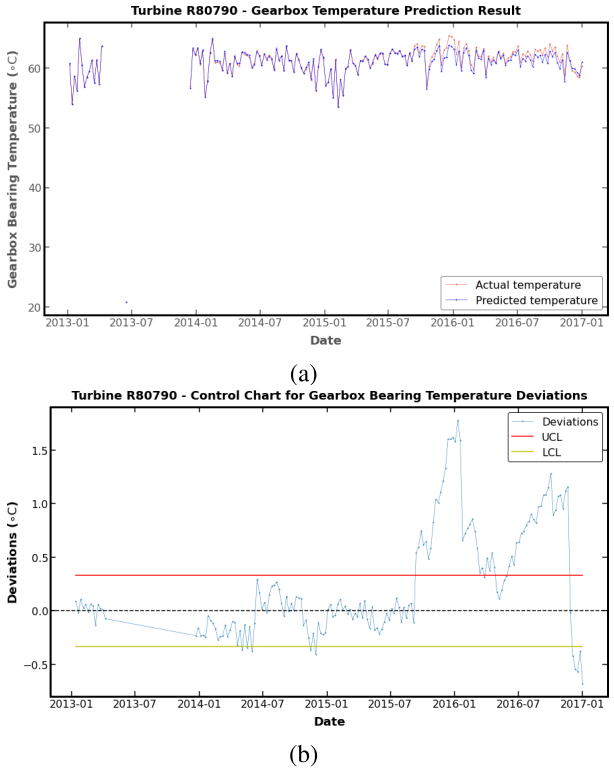


FIGURE 16. WT R80790 Gearbox: (a) Temperature prediction result for generator bearing (b) Control chart for generator bearing temperature deviations.

November 8, 2015, to November 20, 2016, we do not have significant deviations that predicted such massive disruptions.

B. EDP DATASET

1) DATA PREPROCESSING

The data available:

- Historical SCADA data of operation recorded every 10 min from January 1, 2017, to December 31, 2017, for a total of 83 sample variables for four turbines
- Historical Failure Logbook for the year 2017

Some parameters in the SCADA were recorded along with their basic statistics such as minimum, maximum, mean, and standard deviation. Since we have maintenance records in this case study, we must analyse the failure data before selecting an appropriate data set for the training phase. A part of the dataset free from fault was manually selected for the two models to avoid impacting the monitored variables. Although there are no general rules on the ideal size of data to be selected, the chosen dataset must have all the variables (input and output) required to define the normal operation of the WT. Therefore, we decided on a monthly interval of wind turbine(T06) operation and a quarterly for wind turbine T07. Then the stepwise method in the data cleaning subsection of Section III was followed to clean the data. After obtaining the clean data ready to be used for model training, the models' input parameters are extracted from the cleaned SCADA data to construct the input data set. The input data set variables making up the model for each WT component were selected based on the Table.1. In Fig.17 we have the graphical display of the chosen output variables needed to define WT models of its components (i.e., gearbox and generators) across the two turbines in the wind farm in the training phase.

A set of variables required to build each WT component was selected to form an input dataset and the corresponding output variable. The input data set was standardised as discussed in Section III; only after then, we split the entire dataset of the input dataset and the output variable. The data split was done by selecting the first 70% of the data for training and the last 30% for testing. This method of data splitting ensured no data shuffling since the dataset is composed of time series such that any random data selection could result in data leakage.

2) MODEL PROCESSING

The exact process was followed to build the two models, as discussed in Section III.

a: GEARBOX MODEL FOR T06

For WT T06, XGBoost model with $RMSE = 0.7°C$, $MAE = 0.5°C$, $MAPE = 0.8\%$ was chosen because it had the highest R-Squared value; lowest RMSE, MAE and MAPE compared to LSTM and MLR as seen in Table.8.

The XGBoost model used the historical SCADA data obtained by sensors recording the gearbox bearing temperature in WT T06 during normal operation (healthy state) to

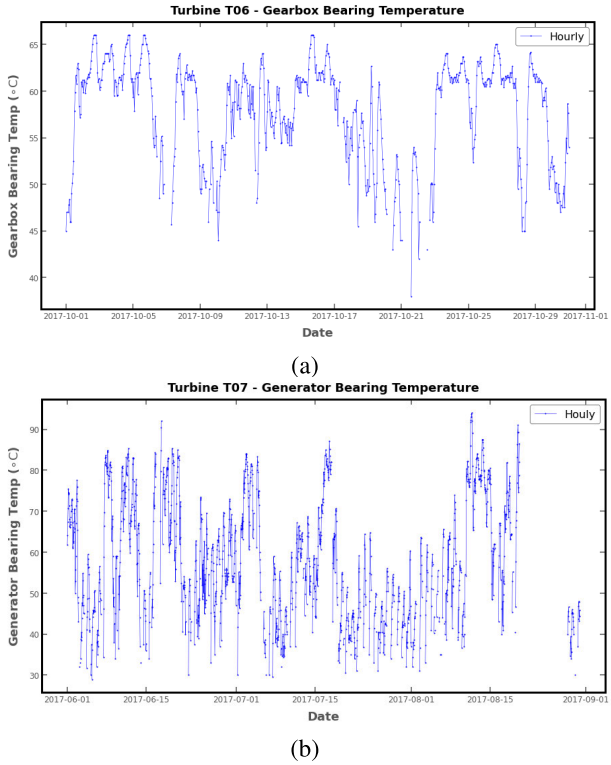


FIGURE 17. Graphical display of selected output variables during training phase (a) Gearbox bearing temperature for WT T06 (b) Generator bearing temperature for WT T07.

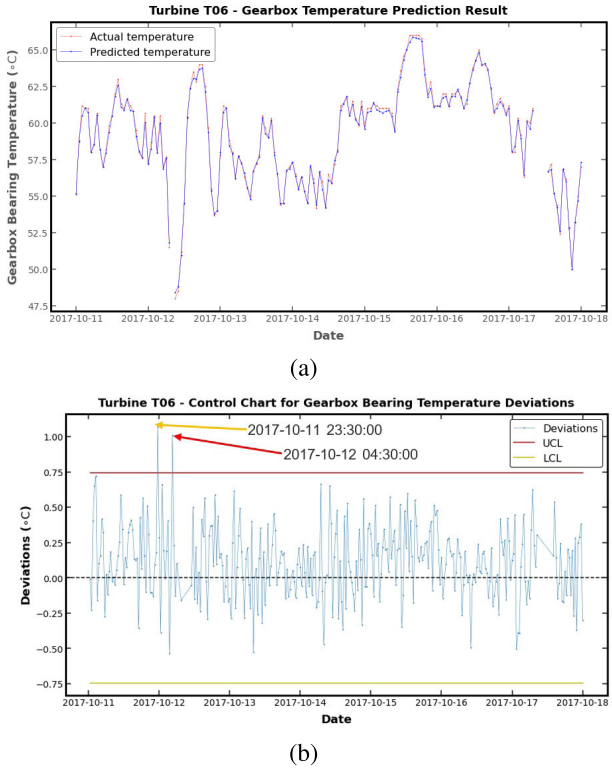


FIGURE 18. WT T06 Gearbox:(a) Temperature prediction result forbearing (b) Control chart for bearing temperature deviations.

TABLE 8. Model accuracy for gearbox bearing temperature prediction.

Turbine	Model	R-Squared	RMSE	MAE	MAPE
T06	XGBoost	0.97712	0.69995	0.50284	0.00865
	LSTM	0.96987	0.80318	0.60619	0.01065
	MLR	0.96704	0.84001	0.66014	0.01131

TABLE 9. Model accuracy for generator bearing temperature prediction.

Turbine	Model	R-Squared	RMSE	MAE	MAPE
T07	XGBoost	0.89976	4.80973	3.81164	0.06492
	LSTM	0.90016	4.80030	3.70982	0.06210
	MLR	0.88610	5.12712	4.33263	0.07486

predict the temperature, as shown in Fig.18(a). From the failure logs, the Gearbox bearings were damaged at timestamp 2017-10-17 08:38. The control chart for this application is represented in Fig.18(b) with a fault threshold of ±0.74°C. We predicted this fault at timestamp 2017-10-11 23:30:00 and a second alarm at timestamp 2017-10-12 04:30:00. The fault detection algorithm showed good predictive ability by alerting of failure six days ahead.

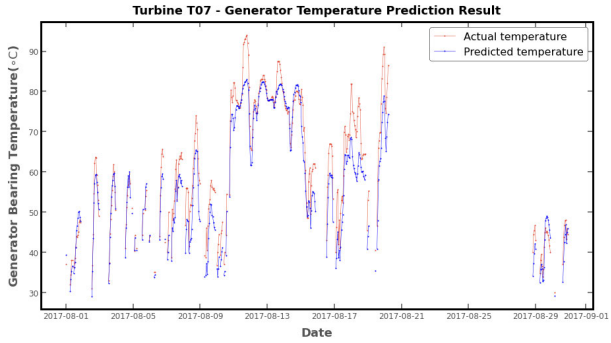
b: GENERATOR MODEL FOR T07

For WT T07, LSTM model with $RMSE = 4.8°C, MAE = 3.71°C, MAPE = 6.2\%$ was chosen because it had the highest R-Squared value, lowest RMSE, MAE, and MAPE compared to XGBoost and MLR as seen in Table9.

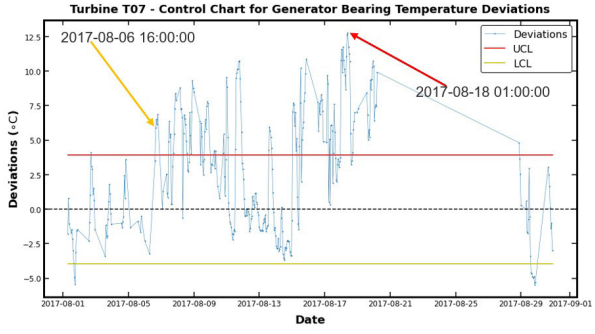
The LSTM model used the historical SCADA data from sensors recording the generator bearing temperature in WT T07 during normal operation (healthy state) to predict the temperature, as shown in Fig.19(a). The control chart for this application is represented in Fig.19(b) with a fault threshold of ±4.07°C. From the failure logs, Generator bearings were damaged at timestamp 2017-08-20 06:08, and the generator was damaged at timestamp 2017-08-21 14:47. We can see from Fig.19(a) that the WT was out of service from August 20, 2017, to August 29, 2017. The catastrophic damage was first predicted by our algorithm at timestamp 2017-08-06 16:00:00, and there were multiple points beyond the fault threshold up to three-time its value at timestamp 2017-08-18 01:00:00 as seen in Fig.19(b). It can be said that the fault detection algorithm was able to predict the generator damage two weeks ahead and gave multiple alarms up to three days before it occurred.

V. DISCUSSION

This study followed three main steps to develop the fault detection algorithm: data acquisition and preprocessing, model processing, and post-processing. The data preprocessing action required a more rigorous process due to the complex working conditions of the WT presented by power restrictions and the presence of outliers in the historical SCADA data. A stepwise approach was followed to eliminate outliers and data points affected by power restrictions to prevent the elimination of valuable data points.



(a)



(b)

FIGURE 19. WT T07 Generator:(a) Temperature prediction result for bearing (b) Control chart for bearing temperature deviations.

The cleaned data was fed into three machine learning algorithms: MLR, XGBoost, and LSTM. The best model was selected based on strict performance metrics using a combination of R-Squared, RMSE, MAE, and MAPE. The selected model was used to predict the output variable required to define WT component normal behaviour (healthy state). Post-processing of the predicted output variable was carried out to determine its deviation from the actual historical record. The sensitivity of these deviations was evaluated using the Shewhart control chart; a fault threshold was established for each model evaluated. Data points outside the fault threshold coupled with a shift in the averages were indicators of a fault in the WT. We presented two case studies using SCADA data from operational wind farms. We gained valuable insights into when the wind turbine will fail even without knowing what failure looks like in the first case study. We validated our approach with the second case study, and our algorithm was able to predict the fault in the WT before the time it occurred, as recorded in the failure logs of the wind farm.

VI. CONCLUSION

In this paper, a system for monitoring and detecting anomalies in the wind turbine gearbox and generator is developed using SCADA data, extreme gradient boosting (XGBoost), and Long Short-Term Memory (LSTM). Statistical Process Control (SPC) is used to evaluate the deviations of predicted signals representing the healthy state of the system and the recorded signals, resulting in fault detection. The proposed method was tested on two real case studies regarding six different WT to determine its effectiveness and applicability. It was observed that the LSTM algorithm outperformed XGBoost in building the generator model for five out of the six WTs, whereas XGBoost better modelled the gearbox. We demonstrated the usefulness of our detection algorithm to detect faults on WT having no failure logs. The fault detection algorithm can assist asset managers of the newly installed wind farms in predicting when the fault will occur and plan for early intervention to prevent catastrophic damage. This system has proven valuable to WT maintenance crew and wind farm asset managers to give a more dynamic data-driven maintenance strategy, which can save the considerable cost of catastrophe failure associated with the current static time-based maintenance strategy. The next step of this paper will be to explore the use of other SPC techniques to explore the sensitivity level of deviations. Also, the use of streaming data to detect the fault and using the deviation signatures from the control chart to carry out fault diagnosis by inferring which specific parts(subcomponents) of the main components are about to fail. This would require working with domain experts to establish data requirements and define the normal behaviour of these subcomponents since we aim to build a robust system that helps the growing wind energy sector optimise and operate cost-effectively.

REFERENCES

[1] *End to Coal Power Brought Forward on October 2024*. Accessed: Jun. 30, 2021. [Online]. Available: https://www.gov.U.K./government/news/end-to-coal-power-brought-forward-to-october-2024

[2] Rystad Energy. *U.K.'s Renewable Energy Capacity Set to Double by 2026, When the Offshore Wind Overtakes Onshore*. Accessed: Oct. 28, 2020. [Online]. Available: https://www.rystadenergy.com/newsevents/news/press-releases/uks-renewable-energy-capacity-set-to-double-by-2026-when-offshore-wind-will-overtake-onshore/

[3] IRENA. (2016). *Wind Energy*. Accessed: Jul. 27, 2021. [Online]. Available: https://www.irena.org/wind

[4] *Renewable Power Generation Costs in 2020*, International Renewable Energy Agency, Abu Dhabi, United Arab Emirates, Jun. 2021.

[5] Z. Xu, J. Wei, S. Zhang, Z. Liu, X. Chen, Q. Yan, and J. Guo, "A state-of-the-art review of the vibration and noise of wind turbine drivetrains," *Sustain. Energy Technol. Assessments*, vol. 48, Dec. 2021, Art. no. 101629, doi: 10.1016/j.seta.2021.101629.

[6] W. Teng, Y. Liu, Y. Huang, L. Song, Y. Liu, and Z. Ma, "Fault detection of planetary subassemblies in a wind turbine gearbox using TQWT based sparse representation," *J. Sound Vib.*, vol. 490, Jan. 2021, Art. no. 115707, doi: 10.1016/j.jsv.2020.115707.

[7] T. Wang, Q. Han, F. Chu, and Z. Feng, "Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review," *Mech. Syst. Signal Process.*, vol. 126, pp. 662–685, Jul. 2019, doi: 10.1016/j.ymssp.2019.02.051.

[8] W. Teng, X. Ding, Y. Zhang, Y. Liu, Z. Ma, and A. Kusiak, "Application of cyclic coherence function to bearing fault detection in a wind turbine generator under electromagnetic vibration," *Mech. Syst. Signal Process.*, vol. 87, pp. 279–293, Mar. 2017, doi: 10.1016/j.ymssp.2016.10.026.

[9] T. MathWorks. *Predictive Maintenance, Part 1: Introduction Video*. Accessed: Aug. 21, 2021. [Online]. Available: https://U.K.mathworks.com/videos/predictive-maintenance-part-1-introduction-1545827554336.html

[10] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, and G. Nenadic, "Machine learning methods for wind turbine condition monitoring: A review," *Renew. Energy*, vol. 133, pp. 620–635, Apr. 2019, doi: 10.1016/j.renene.2018.10.047.

[11] A. F. Dakhil, W. M. Ali, and A. A. Abdulredah, "Predicting prior engine failure with classification algorithms and web-based IoT sensors," in *Proc. Emerg. Technol. Comput., Commun. Electron. (ETCCE)*, Dec. 2020, pp. 1–6, doi: 10.1109/etcce51779.2020.9350895.
[12] K. Leahy, R. L. Hu, I. C. Konstantakopoulos, C. J. Spanos, and A. M. Agogino, "Diagnosing wind turbine faults using machine learning techniques applied to operational data," in *Proc. IEEE Int. Conf. Prognostics Health Manage. (ICPHM)*, 2016, pp. 1–8, doi: 10.1109/ICPHM.2016.7542860.
[13] Y. Liu, Z. Wu, and X. Wang, "Research on fault diagnosis of wind turbine based on SCADA data," *IEEE Access*, vol. 8, pp. 185557–185569, 2020, doi: 10.1109/access.2020.3029435.
[14] Y. Wang, X. Ma, and P. Qian, "Wind turbine fault detection and identification through PCA-based optimal variable selection," *IEEE Trans. Sustain. Energy*, vol. 9, no. 4, pp. 1627–1635, Oct. 2018, doi: 10.1109/tste.2018.2801625.
[15] M. Beretta, J. J. Cárdenas, C. Koch, and J. Cusidó, "Wind fleet generator fault detection via SCADA alarms and autoencoders," *Appl. Sci.*, vol. 10, no. 23, p. 8649, Dec. 2020, doi: 10.3390/app10238649.
[16] A. Verma and A. Kusiak, "Fault monitoring of wind turbine generator brushes: A data-mining approach," *J. Sol. Energy Eng.*, vol. 134, no. 2, pp. 1–5, Feb. 2012, doi: 10.1115/1.4005624.
[17] Z. Liu, C. Xiao, T. Zhang, and X. Zhang, "Research on fault detection for three types of wind turbine subsystems using machine learning," *Energies*, vol. 13, no. 2, p. 460, Jan. 2020, doi: 10.3390/en13020460.
[18] Y. Zhao, D. Li, A. Dong, D. Kang, Q. Lv, and L. Shang, "Fault prediction and diagnosis of wind turbine generators using SCADA data," *Energies*, vol. 10, no. 8, p. 1210, Aug. 2017, doi: 10.3390/en10081210.
[19] J. Chatterjee and N. Dethlefs, "Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines," *Wind Energy*, vol. 23, no. 8, pp. 1693–1710, Aug. 2020, doi: 10.1002/we.2510.
[20] B. Manobel, F. Sehnke, J. A. Lazzás, I. Salfate, M. Felder, and S. Montecinos, "Wind turbine power curve modeling based on Gaussian processes and artificial neural networks," *Renew. Energy*, vol. 125, pp. 1015–1020, Sep. 2018, doi: 10.1016/j.renene.2018.02.081.
[21] M. Schlechtingen, I. F. Santos, and S. Achiche, "Using data-mining approaches for wind turbine power curve monitoring: A comparative study," *IEEE Trans. Sustain. Energy*, vol. 4, no. 3, pp. 671–679, Jul. 2013, doi: 10.1109/tste.2013.2241797.
[22] P. Trizoglou, X. Liu, and Z. Lin, "Fault detection by an ensemble framework of extreme gradient boosting (XGBoost) in the operation of offshore wind turbines," *Renew. Energy*, vol. 179, pp. 945–962, Dec. 2021, doi: 10.1016/j.renene.2021.07.085.
[23] H. Chen, H. Liu, X. Chu, Q. Liu, and D. Xue, "Anomaly detection and critical SCADA parameters identification for wind turbines based on LSTM-AE neural network," *Renew. Energy*, vol. 172, pp. 829–840, Jul. 2021, doi: 10.1016/j.renene.2021.03.078.
[24] J. Fu, J. Chu, P. Guo, and Z. Chen, "Condition monitoring of wind turbine gearbox bearing based on deep learning model," *IEEE Access*, vol. 7, pp. 57078–57087, 2019, doi: 10.1109/access.2019.2912621.
[25] R. Orozco, S. Sheng, C. Phillips, and C. Phillips. (Aug. 2018). *Diagnostic Models for Wind Turbine Gearbox Components Using SCADA Time Series Data*. Accessed: Aug. 6, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8448545.
[26] H. S. Dhiman, D. Deb, J. Carroll, V. Muresan, and M.-L. Unguresan, "Wind turbine gearbox condition monitoring based on class of support vector regression models and residual analysis," *Sensors*, vol. 20, no. 23, p. 6742, Nov. 2020, doi: 10.3390/s20236742.
[27] P. Qian, X. Tian, J. Kanfoud, J. Lee, and T.-H. Gan, "A novel condition monitoring method of wind turbines based on long short-term memory neural network," *Energies*, vol. 12, no. 18, p. 3411, Sep. 2019, doi: 10.3390/en12183411.
[28] F. Castellani, D. Astolfi, and F. Natili, "SCADA data analysis methods for diagnosis of electrical faults to wind turbine generators," *Appl. Sci.*, vol. 11, no. 8, p. 3307, Apr. 2021, doi: 10.3390/app11083307.
[29] A. Santolamazza, D. Dadi, and V. Introna, "A data-mining approach for wind turbine fault detection based on SCADA data analysis using artificial neural networks," *Energies*, vol. 14, no. 7, p. 1845, Mar. 2021, doi: 10.3390/en14071845.
[30] N. M. Khan, G. M. Khan, and P. Matthews, "AI based real-time signal reconstruction for wind farm with SCADA sensor failure," in *Proc. IFIP Adv. Inf. Commun. Technol.*, 2020, pp. 207–218, doi: 10.1007/978-3-030-49186-418.
[31] L.-L. Li, X. Zhao, M.-L. Tseng, and R. R. Tan, "Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm," *J. Cleaner Prod.*, vol. 242, Jan. 2020, Art. no. 118447, doi: 10.1016/j.jclepro.2019.118447.
[32] M. Yesilbudak, "Implementation of novel hybrid approaches for power curve modeling of wind turbines," *Energy Convers. Manage.*, vol. 171, pp. 156–169, Sep. 2018, doi: 10.1016/j.enconman.2018.05.092.
[33] E. Renewables. (Oct. 9, 2019). *La Haute Borne Data*. Accessed: Aug. 14, 2021.[Online]. Available: https://opendata-renewables.engie.com/explore/dataset/d543716b-368d-4c53-8fb1-55addbe8d3ad/information
[34] L. Ziegler, E. Gonzalez, T. Rubert, U. Smolka, and J. J. Melero, "Lifetime extension of onshore wind turbines: A review covering Germany, Spain, Denmark, and the UK," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 1261–1271, Feb. 2018, doi: 10.1016/j.rser.2017.09.100.
[35] T. Chen and C. Guestrin, "XGBoost," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2016, doi: 10.1145/2939672.2939785.
[36] Y. Wang, S. Sun, X. Chen, X. Zeng, Y. Kong, J. Chen, Y. Guo, and T. Wang, "Short-term load forecasting of industrial customers based on SVMD and XGBoost," *Int. J. Electr. Power Energy Syst.*, vol. 129, Jul. 2021, Art. no. 106830, doi: 10.1016/j.ijepes.2021.106830.
[37] P. Guo and D. Infield, "Wind turbine blade icing detection with multi-model collaborative monitoring method," *Renew. Energy*, vol. 179, pp. 1098–1105, Dec. 2021, doi: 10.1016/j.renene.2021.07.120.
[38] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018, doi: 10.1109/access.2018.2818678.
[39] P. C. Bhat, H. B. Prosper, S. Sekmen, and C. Stewart, "Optimisingg event selection with the random grid search," *Comput. Phys. Commun.*, vol. 228, pp. 245–257, Jul. 2018, doi: 10.1016/j.cpc.2018.02.018.
[40] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
[42] C. Olah. (2015). *Understanding LSTM Networks–Colah's Blog*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[43] E. Inovação. (Aug. 8, 2018). *Explore-EDPOpenData*. Accessed: Aug. 14, 2021. [Online]. Available: https://opendata.edp.com/explore/?refine.keyword=visible&sort=modified
[44] D. Menezes, M. Mendes, J. A. Almeida, and T. Farinha, "Wind farm and resource datasets: A comprehensive survey and overview," *Energies*, vol. 13, no. 18, p. 4702, Sep. 2020, doi: 10.3390/en13184702.

WISDOM UDO received the B.Eng. degree in mechanical (production) engineering from the University of Benin, Nigeria, in 2014. He is currently pursuing the master's degree in engineering and computing with Teesside University, U.K.

His research interests include the reliability of wind turbines and rotating machines.

YAR MUHAMMAD (Senior Member, IEEE) received the master's degree in computer engineering from Mid Sweden University, in 2009, and the Ph.D. degree in information communication technology (ICT) from the Tallinn University of Technology, in 2015.

He taught at the University of Tartu. He is currently working as a Senior Lecturer (Assistant Professor) with Teesside University, U.K., where he is also a part of the Centre for Digital Innovation. He received the Young Investigator Award, which Springer and IFMBE awarded at 16th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics and Medicinteknikdagarna 2014, Sweden; and he was runner-up for the Best Paper Award in the 26th ISSC, Ireland, in 2015.